JP-LINK： セキュリティ・サーバのアーキテクチャ

1 はじめに

このドキュメントでは、JP-LINK セキュリティ・サーバのアーキテクチャについて説明する。 JP-LINK とセキュリティ・サーバの役割の詳細については、[JP-LINK の技術仕様]を参照してください。
このドキュメントでは、セキュリティ・サーバのコンポーネントの概要と、これらのコンポーネントの間のインターフェイスについて説明する。セキュリティ・サーバの内部動作の概要を取得したい技術者を対象とする。
セキュリティ・サーバのコンポーネントの技術仕様のみを提供する。これらのコンポーネント間の相互運用性の具体的な例については、別のドキュメントを参照してください。

1.1 概要

セキュリティ・サーバの主な機能は、サービス・クライアントとサービス・プロバイダの間のリクエストを仲介することである。クライアントとプロバイダの両方は、情報システムに接続しているセキュリティ・サーバと通信し、二者間のセキュアなメッセージ交換はセキュリティ・サーバによって処理される。
パブリックインターネットを介して送信されるメッセージは、デジタル署名と暗号化によって保護される。
サービス・プロバイダのセキュリティ・サーバは受信するメッセージにアクセス制御を適用し、それによって、サービス・プロバイダと適切な契約を結んだユーザーのみがデータにアクセスできることを確保する。

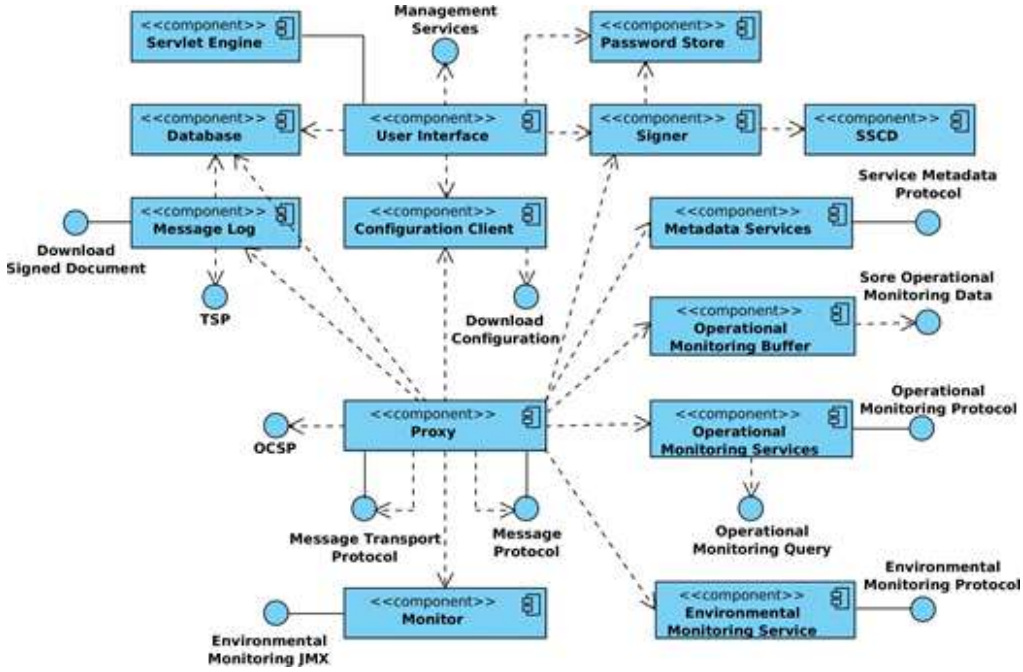
セキュリティ・サーバは、グローバル・コンフィギュレーションを提供するセントラル・サーバにも依存する。

1.2 用語と定義

MIME - Multipurpose Internet Mail Extensions
OCSP - Online Certificate Status Protocol
SOAP - Simple Object Access Protocol
TLS - Transport Layer Security
TSP - Time Stamp Provider

2　コンポーネント・ビュー

図 1 は、JP-LINK セキュリティ・サーバの主要なコンポーネントとインターフェイスを示す。コンポーネントとインターフェイスについては、以降のセクションで詳しく説明する。



2.1　プロキシ

プロキシは、サービス・クライアントとサービス・プロバイダの間のメッセージを仲介する。メッセージはパブリックインターネットを介して送信され、プロキシは、通信がデジタル署名と暗号化により保護されていることを保証する。

このコンポーネントはスタンドアローンの Java デーモン・アプリケーションである。

2.2　メッセージ・ログ

セキュリティ・サーバを通過するすべての通常のメッセージをデータベースに記録する。メッセージは署名とともに保存され、署名にはタイムスタンプが発行される。メッセージ・ログの目的は、リクエスト/レスポンスメッセージの受信を第三者に証明する手段を提供することである。

データベースのログ・レコードを署名付き文書として定期的にファイルに書き出し、アーカイブされたログ・レコードをデータベースから削除する。格納された情報を含む署名付き文書のデータベースからの取得を可能にするサービスを提供する。

このコンポーネントはプロキシへのアドオンである。

2.3　メタデータ・サービス

JP-LINK 参加者がセキュリティ・サーバから利用可能なサービスを検出するための方法を提供する。

2.4　運用モニタリング・サービス

JP-LINK 参加者がセキュリティ・サーバの運用モニタリング情報を取得する方法を提供する。
このコンポーネントはプロキシへのアドオンである。

2.5 運用モニタリング・バッファ

運用モニタリング・バッファは、プロキシの運用モニタリング・データを、運用モニタリング・デーモンに仲介するメモリ内循環バッファである。このコンポーネントはプロキシへのアドオンである。

2.6 Signer

Signer コンポーネントは、メッセージの署名用鍵と証明書を管理する。Signer は、メッセージに署名する時に、およびその有効性を検証する時に、プロキシ・コンポーネントから呼び出される。また、ユーザー・インターフェイスは、認証の作成、鍵の署名、およびリクエストの証明する時にも Signer を呼び出す。

このコンポーネントはスタンドアローンの Java デーモン・アプリケーションである。

2.7 データベース

セキュリティ・サーバの設定は、PostgreSQL データベースに保持される。セキュリティ・サーバの設定の詳細については、別ドキュメントを参照してください。設定は、セキュリティ・サーバのユーザー・インターフェイスを介して変更することがでる。

2.8 ユーザー・インターフェイス

ユーザーは、セキュリティ・サーバのユーザー・インターフェイスを介して、セキュリティ・サーバの設定を管理できる。ユーザー・インターフェイスは、Web ベースで、サーブレット・エンジンにデプロイされる war アーカイブとしてパッケージ化されている。
特定の動作は、JP-LINK グローバル・コンフィギュレーションを変更することを試みて、管理リクエストを JP-Link センター・サーバに送信する場合がある。このリクエストは、グローバル・コンフィギュレーションに反映される前に、セントラル・サーバ管理者によって承認される必要がある。

システム状態または設定を変更するユーザー・アクション・イベントは、監査ログに記録される。全てのアクションは、結果が成功か失敗かにかかわらず記録される。監査ログ・イベントの完全なリストは、別ドキュメントに記述されている。

2.9 サーブレット・エンジン

Jetty サーブレット・エンジンは、ユーザー・インターフェイスをホストし、JP-LINK コンフィギュレーション・ファイルの中で設定可能なポートを使ってリスニングしている。

2.10 コンフィギュレーション・クライアント

コンフィギュレーション・クライアントは、リモート・グローバル・コンフィギュレーション・ファイルをダウンロードする。グローバル・コンフィギュレーションのソース・ロケーションは、セキュリティ・サーバのユーザー・インターフェイスからアップロードされたアンカー・ファイルから取得される。

このコンポーネントはスタンドアローンの Java デーモン・アプリケーションである。

2.11 パスワード・ストア

セキュリティ・トークンのパスワードを、セキュリティ・サーバのインターフェイスと Signer がアクセスできるオペレーティング・システムの共有メモリ・セグメントに格納される。セキュリティ・サーバが再起動するまで、パスワードの漏洩することなくセキュリティ・トークンのログイン状態を維持する。

2.12 SSCD

SCD(セキュア著名デバイス)は、Signerにセキュアな暗号署名作成機能を提供するオプショナル・ハードウェア・コンポーネントである。
SSCD は、交換するメッセージを署名するためにセキュリティ・サーバが任意に追加できる PKCS #11([PKCS11])準拠のハードウェア・デバイスである必要がある。このインターフェイスを使用するためには、セキュリティ・サーバ・システムに PKCS #11 対応デバイス・ドライ

バをインストールおよび設定する必要がある。

2.13 環境モニタリング・サービス

X-Road 参加者がセキュリティ・サーバの環境データを取得する方法を提供する。Akka インターフェイスを介してローカル・モニタリング・サービスにデータをリクエストし、それを SOAP XML レスポンスに変換する。

このコンポーネントはプロキシへのアドオンである。

2.14 モニター

モニター・コンポーネントは、実行中のプロセス、使用可能なディスク容量、インストールされたパッケージなどの環境モニタリング情報を収集する。モニタリング・データは、Akka および JMX インターフェイスを介して公開される。

3 インターフェイス

3.1 管理サービス

管理サービスはセキュリティ・サーバによって呼び出され、セキュリティ・サーバ・クライアントの登録や認証用証明書の削除などの管理タスクを実行する。

管理サービス・インターフェイスは、セキュリティ・サーバによって要求される同期 RPC スタイルインターフェイスである。サービスはセントラル・サーバによって提供される。

このインターフェイスは、別ドキュメントでより詳細に説明する。

3.2 コンフィギュレーションのダウンロード

セキュリティ・サーバは、作成されたグローバル・コンフィギュレーション・ファイルをコンフィギュレーション・ソースからダウンロードする。

コンフィギュレーション・ダウンロード・インタフェースは、セキュリティ・サーバによって要求される同期インターフェイスである。これは、セントラル・サーバやコンフィギュレーション・プロキシなどのコンフィギュレーション・ソースによって提供される。

このインターフェイスについては、別ドキュメントで詳しく説明する。

3.3 メッセージ・プロトコル

X-Road メッセージ・プロトコルは、X-Road セキュリティ・サーバと通信するために、サービス・クライアントおよびサービス・プロバイダの情報システムが使用する。

このプロトコルは、クライアント情報システムまたはサービス・プロバイダのセキュリティ・サーバによって開始される同期 RPC プロトコルである。

3.4 メッセージ転送プロトコル

X-Road メッセージ転送プロトコルは、サービス・リクエストとサービス・レスポンスを交換するためにセキュリティ・サーバが使用する。このプロトコルは、サービス・クライアントのセキュリティ・サーバによって開始される同期 RPC プロトコルである。

このインターフェイスについては、別ドキュメントで詳しく説明する。

3.5 サービス・メタデータ・プロトコル

X-Road サービス・メタデータ・プロトコルは、サービス・クライアントの情報システムが、X-Road インスタンスに関する情報を収集するために使用する。特に、このプロトコルは、X-Road メンバー、これらのメンバーによって提供されるサービス、および WSDL サービス記述を発見するために使用できる。
このプロトコルは、サービス・クライアント情報システムによって開始される同期 RPC プロトコルである。 このインターフェイスについては、別ドキュメントで詳しく説明する。

3.6 署名された文書のダウンロード

署名された文書をダウンロードするサービスは、セキュリティ・サーバのメッセージ・ログから署名されたコンテナをダウンロードするために、情報システムによって使用される。コンテナは第三者に転送し、オフラインで確認することができる。さらに、このサービスは、署名されたコンテナを検証するために使用するグローバル・コンフィギュレーションをダウンロードする簡易な方法を提供する。

このプロトコルは、情報システムによって開始される同期 RPC スタイルのプロトコルである。このインターフェイスについては、別ドキュメントで詳しく説明する。

3.7 TSP

タイムスタンプ・プロトコル(TSP)は、交換されたメッセージの長期的な証拠とするためにセキュリティ・サーバに使用される。セキュリティ・サーバは、すべて　　のメッセージとその署名を記録する。これらのログに長期的な証明のために、定期的にタイムスタンプが発行されている。

タイムスタンプ・プロトコルは、タイムスタンプ局によって提供される同期プロトコルである。 このインターフェイスについては、[X-Road の技術仕様]で詳しく説明する。

3.8 OCSP

OCSP プロトコルは、セキュリティ・サーバが署名と認証用証明書に関する有効性情報を検索するために使用する。OCSP プロトコルは、認証局に属する OCSP レスポンダによって提供される同期プロトコルである。

このインターフェイスについては、[X-Road の技術仕様]で詳しく説明する。

3.9 運用モニタリング・プロトコル

X-Road 運用モニタリング・プロトコルは、外部モニタリング・システムによって、セキュリティ・サーバの運用情報を収集するために使用される。このプロトコルは、外部モニタリング・システムによって開始される同期 RPC プロトコルである。

The protocol is described in more detail in [PR-OPMON].

このインターフェイスについては、別ドキュメントで詳しく説明する。

3.10 運用モニタリング・データの保存

セキュリティ・サーバは、このプロトコルを使用して、運用モニター・データを運用モニター・デーモンに保管する。このプロトコルは、外部モニタリング・シス　　テムによって開始される同期 RPC プロトコルである。

このインターフェイスについては、別ドキュメントで詳しく説明する。

3.11 運用モニタリング・クエリ

セキュリティ・サーバは、このプロトコルを通じて、運用モニター・リクエストを稼動モニター・デーモンに仲介する。このプロトコルは、外部モニタリング・システムによって開始される同期 RPC プロトコルである。

このインターフェイスについては、別ドキュメントで詳しく説明する。

3.12 環境モニタリング・プロトコル

環境モニタリング・プロトコルは、外部モニタリング・システムによって、セキュリティ・サーバに関する環境モニタリング情報を収集することに使用される。

3.13 環境モニタリング JMX

Monitor JMX インタフェーイスは、環境モニタリングサービスによって収集されたローカル・セキュリティ・サーバの環境モニタリング・データを公開する。

4　テクノロジー・マトリックス

表 1 は、セキュリティ・サーバで使用されるテクノロジーの一覧と、テクノロジーとセキュリティ・サーバ・コンポーネント間のマッピングを示す。

Technology	Signer	Proxy	Password Store	Message Log	Metadata Services	Database	Configuration Client	User Interface	Servlet Engine	Monitor	Environ Monitor Service
Java 8	✓	✓		✓	✓		✓	✓	✓	✓	✓
C			✓								
Logback	✓	✓		✓	✓		✓	✓			✓
Akka 2.X	✓	✓		✓				✓		✓	✓
Jetty 9									✓		
JRuby 1.7								✓			
Javascript								✓			
PostgreSQL 9.3						✓					
PAM									✓		
Liquibase						✓					
upstart		✓	✓				✓		✓		
PKCS #11[3]		✓									
Dropwizard Metrics										✓	

表 1.セキュリティ・サーバのテクノロジー・マトリックス

[3]暗号ハードウェア・デバイスを使用するためには、PKCS #11 対応のデバイス・ドライバをシステムにインストール、設定する必要がある。

5 配置ビュー

5.1 シンプルな配置

可用性が重大な問題ではないシナリオ(環境のテスティングなど)の場合、一台のセキュリティ・サーバを使用できる。認証用鍵と署名用鍵は、HSM デバイスに格納される。図 2 に、対応する配置図を示す。

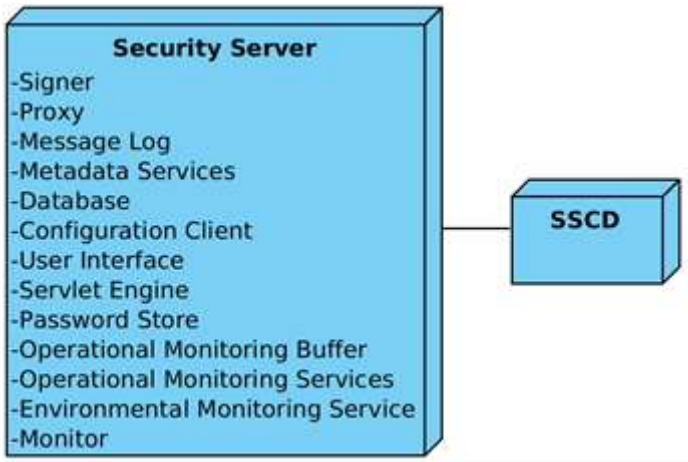


図 2 一台のセキュリティ・サーバの配置

暗号化ハードウェア・デバイスによってメッセージの署名が提供される場合には、任意で SSCD をセキュリティ・サーバに接続することができる。

5.2 配置の冗長化

冗長化によって、システム全体の可用性を高めることができる。情報システムは、負荷を均等に分散するために、ロード・バランサを介して複数台のセキュリティ・サーバに接続することができる。複数台のセキュリティ・サーバが、サービス・クライアントにリクエストされるサービスを提供する場合、サービス・クライアントのセキュリティ・サーバは、リクエストを転送するときに最初の利用可能なサービス・プロバイダのセキュリティ・サーバを選択する。したがって、同じサービスを提供するように設定された複数台のセキュリティ・サーバが存在する場合、X-Road メッセージ転送プロトコルには冗長性がある。